Dublin City University

Electronic and Computer Engineering

Streaming Audio Server with Listener-Tracking Embedded Clients - Status Report

Authors

Michael Lenehan michael.lenehan4@mail.dcu.ie

Supervisor

Martin Collier martin.collier@mail.dcu.ie

December 3, 2018

Table of Contents

1 Introduction

There are many options for open-source audio streaming available to users. While subscription based options such as Spotify exist, and there are options for the playback of locally stored media, an optimal solution would serve both options to listeners, to any connected device.

This project explores the idea of implementing a streaming server which allows users access a stored collection of audio, from any connected device, and to stream this audio to the nearest available client device. A solution must offer an accessible user experience, and importantly provide good quality playback.

1.1 Project Aim and Scope

The aim of this project is to investigate the available open-source music streaming hardware and software, and to analyse their performance. Metrics which must be analysed include server and client power consumption, CPU and network utilization, and additional required hardware. The usability and expandability of each software package must also be analysed.

Once the optimal software solution is chosen, a listener-tracking implementation can be added. This must allow the server to serve audio to the client device nearest to the listener. Tracking may be integrated using a number of methods, such as Bluetooth proximity sensing, or WiFi positioning using Round-Trip Time, or Angle-of-Arrival.

While this project has a strong focus on the audio server software, there must also be a focus on the hardware which runs this software. The capability of the hardware to serve audio continuously to a user, with minimal artefacts, or degradation of the source audio, is crucial to the project. As there are many embedded Linux based single-board computers available, the performance of each must be compared in order to decide on the system most appropriate for the purpose of streaming audio.

The outcome of the project is to present an open-source alternative to locally stored music libraries, which serves stored or streamed audio to a user device, or to a nearby client output, as determined by WiFi tracking or Bluetooth proximity sensing.

2 Background Information

2.1 Open Source Software

A number of software solutions exist for streaming audio from low powered hardware. Options such as “mpd” - the Music Player Daemon -, mopidy, and volumio, allow for users to play music on the system. These options are typically used to implement headless audio player setups, with the user sending messages over the network to control the player.

While these options provide much of the basic required functionality, they are not a suitable solution for the project. The required functionality from the software will be to stream media from the server system to the client system. This functionality exists in these open-source software options, but requires modifications to be made to configuration files in order to be implemented.

Audio software PulseAudio may also be utilised, as it is often found on UNIX based systems. PulseAudio is a sound server which routes audio from the running application to the selected output device. On Linux systems, this is used to send audio output to the system speakers, or connected USB devices. However, the functionality exists to pass the output audio over the network to a specified address[1].

2.2 Hardware Requirements

There are a number of Linux based embedded systems which may be used to act as a streaming server. Commonly used systems include the Raspberry Pi, BeagleBone Black, and ASUS Tinker Board. There are differences between these development platforms which allow them more or less suitability for the purposes of this project.

The BeagleBone Black (BBB) is a low cost platform, with compatibility for many Linux distributions. The device has on board flash memory, Ethernet and HDMI outputs. There is also on board I^2S support, allowing for hardware Digital to Analog Converters (DACs) to be connected. The BBB has 512MB of DDR3 RAM, and a 1GHz ARM processor on board[2].

The ASUS Tinker Board is a small form-factor Single Board Computer (SBC). The computer has Gigabit Ethernet, HDMI output, multiple IO, including 40 GPIO pins and 4 USB ports. The 1.8GHz ARM based CPU provides high performance when coupled with the 600MHz GPU and 2GB of dual-channel DDR3 RAM. This SBC also supports the I^2S audio protocol[3].

The Raspberry Pi 3 Model B+ is one of the most commonly used embedded Linux development platforms. The device has a 1.4GHz ARM processor, 1GB of DDR2 RAM, Gigabit Ethernet, Bluetooth Low Energy, and multiple IO ports. Again, this board supports the I^2S protocol, with outputs on its GPIO[4].

Each of the aforementioned options offers different levels of performance at different price points. The BeagleBone Black is both the cheapest and least powerful option. The ASUS TinkerBoard is the most powerful and most expensive option, while the Raspberry Pi offers comparatively high performance at a mid price. The benefits and costs of these Single Board Computers must be compared in order to choose that which is most appropriate for the application of serving and streaming audio.

3 Problems and Solutions

There are a number of problems which must be addressed in completing this project. With all of the audio servers listed, the intended use case is a controllable, stationary, headless playback system. As such, modifications must be done to allow the server to send music across a network to a client device. This may be done in a number of ways, for example, in mpd, the configuration file may be modified to allow http streaming[5]. Mopidy also allows for streaming through the modification of the configuration file[Mopidy18]. Volumio requires multiple files to be modified in order to allow it stream. This is done by utilising the PulseAudio audio server in order to route the audio from the server to the sink device[6].

The required software solution must be extensible, allowing users to create add functionality as needed. This will allow for features such as audio routing to be implemented with the final project. It will also allow for streaming of music that is not locally stored, such as from services like Spotify. The Spotify API provides instruction on implementing certain playback functionalities into applications[7].

As one aspect of the project is to analyse the performance of the software, in terms of throughput, packet loss, and jitter. Software packages such as Wireshark allow for analysis of packets transmit on a network. It has the documented intended uses of allowing "troubleshooting network problems", "debugging of protocol implementations", and "verifying network applications"[8]. Using Wireshark, the packets sent from the server to the client may be analysed and the required information may then be extracted.

As previously mentioned, there are multiple available options for Single Board Computers which can be used to run the Linux server. These options range in price and performance. However, due to compatibility with the required software, and the availability of information, the Raspberry Pi 3 Model B+ was chosen. While cheaper and less powerful than the ASUS Tinker Board, it is supported by all software required for this project. The hardware specifications of the board are sufficient for the streaming of high quality audio, and may be used at a further stage of the project to act as client devices with their I^2S outputs providing high quality audio through an external DAC.

Another aspect of the project is the client tracking and audio routing. There are multiple protocols which may be used in order to determine location of a mobile device. Bluetooth "Beacons" or Access Points are used, sending a packet to the device, the signal strength may be used to calculation and approximate distance[9]. Using filtering techniques, distances may be calculated, using Bluetooth, to an accuracy of approximately 1.8 meters[9].

Location using WiFi may provide greater accuracy, however may require specialised hardware[10]. The 802.11ac WiFi standard allows for the use of Beamforming, in which multiple antennas transmit at once, allowing for the targeted transmission of data[11]. Using Angle-of-Arrival (AoA), and Time-of-Flight (ToF), and Multiple Signal Classification (MUSIC) algorithms, the distance and direction from the Access Point may be determined [12]. The operations which must be performed are complex, and dependent on the hardware being used. As such, less complex solutions, such as RSSI, may be implemented, however there is also a reduction in accuracy.

4 Project Plan

As the project spans two college semesters, the workload has been divided across the semesters. While the first college semester has more modules in the first semester, less time is available to work on the project. As such, the work done on the project in this semester will mainly be research focused, with information being gathered which may be helpful in furthering the progress of the project. The second semester contains less taught module hours, and as such, much of the project will be completed in this semester. The plans laid out below are non-exhaustive, not listing any work to be completed between semesters, and do not account for time spent on the project report, or concurrency in completing tasks.

4.1 Semester 1 Plan

The plan for Semester 1 is as follows:

Table 4.1: Semester 1 Plan

Week Number	Tasks
4	Investigated Project Aim
5	Read Audio Server Software Documentation
6	Investigated Android Development Process
7	Completed Android Studio “Kotlin” Introduction Course
8	Installed MPD, Mopidy and Volumio for testing
9	Investigate Network Analysis Software
10	Modify Software Configuration Files to Implement Audio Streaming
11	Investigate Hardware Performance while Streaming Audio
12	Begin Android Client App Development

Much of the basis of the project is planned to be completed by the conclusion of this semester. This includes gathering data for the hardware performance while the server is streaming audio, and the initial design of the Android based client application.

4.2 Semester 2 Plan

The plan for Semester 2 is as follows:

Table 4.2: Semester 2 Plan

Week Number	Tasks
1	Begin Working on Location Tracking
2	Continue Work on Android Client Application
3	Compare Network Performance of Streaming Servers
4	Compare Hardware Performance while Streaming
5	Work on Embedded Playback Devices
6	Implement Audio Routing to Client Playback Devices
7	Work on I^2S DAC Implementation
8	Test Hardware Performance with Implemented DAC
9	Test Tracking with Audio Routing
10	Complete Project Code and Report

A majority of the work on the project is planned for completion in this semester. The location tracking will be worked on alongside the client side application, testing multiple methods for performance and accuracy, The gathered hardware and network performance data will be compared, and the hardware clients, which will consist of a SBC and a DAC, will also be designed and implemented. The deadline for the project will be Week 11 of Semester 2, and, as such, the project must be completed before or within Week 10.

5 Appendix

5.1 Additional Information

Additional research performed has yielded the following results for location tracking implementations:

Location Tracking Sources

Decimeter Level Localization using WiFi Study[13].

Basic information about the 802.11ac protocol[14].

A Study on the performance of OTS networking hardware for positioning purposes[15].

A GitHub Repository for the Find3 indoor positioning framework[16].

Raspberry Pi RSSI assessment for WSN[17].

Indoor Client Centered Positioning with the 802.11ac standard[18].

Single Access Point solution for Location Tracking[19].

Single Access Point solution for Location Tracking[20].

Google Documentation for Android RTT implementation[21].

Bibliography

[1] "PulseAudio," *https://www.freedesktop.org/wiki/Software/PulseAudio/About/*. 2014.

[2] BeagleBoard.org, "BeagleBone black," *https://beagleboard.org/black*. 2018.

[3] ASUS, "ASUS tinker board," *https://www.asus.com/ie/Single-Board-Computer/Tinker-Board/*..

[4] "Raspberry Pi Foundation", "Raspberry pi 3 model b+," *https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/*..

[5] "The Music Player Daemon Project", "Music player daemon 0.21.4 documentation," *https://www.musicpd.org/doc/*. 2018.

[6] Volumio, "Volumio documentation," *https://volumio.github.io/docs/index.html*. 2018.

[7] Spotify, "SpotifyforDevelopers," *https://developer.spotify.com/*. 2018.

[8] Wireshark.org, "Wireshark user's guide," *https://www.wireshark.org/docs/wsug_html_chunked/*..

[9] J. Park, J. Kim, and S. Kang, "BLE-based accurate indoor location tracking for home and office," *Computer Science & Information Technology*, 2015.

[10] J. Gjengset, J. Xiong, G. McPhilips, and K. Jamieson, "Phaser: Enabling phased array signal processing on commodity wifi access points," *School of Information Systems at Institutional Knowledge at Singapore Management University*, 2014.

[11] Y. Heejung and T. Kim, "Beamforming transmission in ieee 802.1ac under time-varying channels," *The Scientific World Journal*, 2014.

[12] A. Uddin Afaz, R. Arablouei, F. de Hoog, B. Kusy, R. Jurdak, and N. Bergmann, "Estimating angle-of-arrival and time-of-flight for multipath components using wifi channel state information," *Sensors*, 2018.

[13] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "SpotFi: Decimeter level localization using wifi," *stanford.edu*, 2016.

[14] M. S. Gast, "802.11ac: A survival guide," *https://www.oreilly.com/library/view/80211ac-*

a-survival/9781449357702/ch04.html..

[15] Y. Chen, "Evaluating off-the-shelf hardware for indoor positioning," *Department of Electrical and Information Technology, Lund University*, 2017.

[16] Z. Scholl, V. Hooi, N. Soni, A. Reed, J. Gaussen, M. Adams, Linder William, "howlofthesun", and "notoriousbdg", "Find3 github repository," *https://github.com/schollz/find3*. 2018.

[17] M. Jais, T. Sabapathy, M. Jusoh, P. Ehkan, L. Murukesan, I. Ismail, and R. Ahmad, "Received signal strength indication (rssi) code assessment for wireless sensors network (wsn) deployed raspberry-pi," *2015 International Conference on Robotics, Automation and Sciences (ICORAS)*. 2015.

[18] W.-S. Jung and Y.-B. Ko, "IEEE 802.11ac based client centered indoor positioning system," *2016 IEEE Canadian Conference on Electrical and Computer Engineering*. 2016.

[19] A. Mariakakis, S. Souvik, J. Lee, and K.-H. Kim, "SAIL: Single access point-based indoor localization," 2014.

[20] V. Romanov and G. Succi, "WLAN based positioning with a single access point," *International Journal of Wireless & Mobile Networks*, 2018.

[21] Google, "Wi-fi location: Ranging with rtt," *https://developer.android.com/guide/topics/connectivity/w rtt*. 2018.